

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

End of Result Set

☐ [Generate Collection](#) [Print](#)

L18: Entry 1 of 1

File: USPT

Apr 9, 2002

DOCUMENT-IDENTIFIER: US 6370513 B1

TITLE: Method and apparatus for automated selection, organization, and recommendation of items

Brief Summary Text (6):

As another example, individuals are often faced with a decision as to what to cook for dinner. In today's environment most families consist of either two working spouses or a single working parent. As a consequence, at least one parent must come home from a long day at work and tackle the decision of what to make for dinner. Since the parent usually gets home from work around dinner time, the children and other family members are usually already hungry and tired. This situation often puts much stress on family members, in particular, the parent tasked with determining what meal to prepare (the primary care providing parent). The primary care providing parent is often pressured to prepare something quickly. Under this pressure, the parent opts most of the time for something simple and quick--if they have all the ingredients that they need to prepare the meal. If the parent does not have all the ingredients, he or she must make a trip to the store or figure out what they can make out of the ingredients at home. As a consequence of this situation, the family's menu is unimaginative. The family will therefore typically end up eating the same food over and over again, or the family will eat food from the same outside vendor (take-out).

Drawing Description Text (24):

FIG. 22 is an exemplary GUI for displaying a weekly menu according to one embodiment of the present invention;

Drawing Description Text (29):

FIG. 27 is an exemplary GUI for the menu addition servlet of FIG. 15; and

Detailed Description Text (46):

For certain items, it may be desirable to keep track of the recommended items actually sampled by the user, as well as when the sampling took place. For instance, in a recipe recommendation system, it may be desirable to keep a time factor attached to each sampled recipe to prevent recipes from being recommended too often. This helps to add variety to a user's menu. According to one embodiment of the invention, the time factor changes from zero to a value between zero and one. The system remembers when an item was last used by storing the date the item was sampled. The system also has a standard number of days after which an item can be used again. If the item has not been used within this time, the time factor for the item is set to one. Otherwise, the time factor is set to a fraction of one based on the number of days it was last used. For example, for a system where the standard number of days is set to ten, an item that has not been used in the past ten days will have one as its time factor. On the other hand, an item that was used only five days ago will have its time factor set to 0.5. According to this embodiment, the program asks in step 460 whether there are any more items in the sorted list of suitable items on which to perform a time factor calculation. If there are, the program, in step 470, multiplies the suitability weight of a current item in the list with a corresponding time factor. This process continues until the suitability weights of all the items in the sorted list have been recomputed based on their time factors. The list is then rearranged in step 480 based on the new calculations.

Detailed Description Text (47):

In step 490, the program recommends the top items from the sorted list. The actual number of items recommended is predetermined by either the user or systems programmer. According to one embodiment of the invention, the top seven items are recommended as the menu for the week.

Detailed Description Text (86):

Recipe vectors are created with the aid of a parser 874 which is in communication with an original recipes directory 876 and a chemical database 878. The parser 874 takes a recipe in the original recipes directory 876, parses out the ingredients in the recipe, and maps the ingredients to the chemicals in the chemical database 878, as is discussed in further detail below. A parsed recipe may be modified via a menu addition servlet 880. The servlet, moreover, allows recipes to be entered directly into the recipe database 872 without invoking the parser.

Detailed Description Text (89):

If the program does not find an ingredient in the chemical database, a substitute ingredient that is found in the database is used in its place. The substitution may be done manually by a systems programmer via the menu addition servlet 880. Alternatively, the system might select a key phrase in the ingredient's name, and find an ingredient in the recipe database that includes the selected key phrase.

Detailed Description Text (95):

The system further allows the user to rate up to five other dishes that the user likes 904 to get better knowledge of the user's tastes. The user accesses a list of dishes stored in the recipes database 872 by selecting a down-arrow button 906, and further selecting a dish from a resultant pull-down menu of dishes.

Detailed Description Text (102):

A user may view his or her weekly menu by selecting a weekly menu option 914, as illustrated in FIG. 22. The user may also view recommendations for a different number of days (e.g. the next two weeks) by entering a desired number in a "Number of days" 916 field, and selecting a "find dishes" button 924. For a day specified as a cook-in day, the system displays recommendations of specified type of dishes (e.g. soup, salad, entree, etc.). A "Show Recipe" button 924 next to the recommended dish allows the user to view a picture of the prepared dish, the dish ingredients, and preparation instructions.

Detailed Description Text (108):

If a restaurant is to be recommended to a group of individuals, the system calculates the vector distance to the dishes of a particular restaurant, and calculates the average vector distance for that restaurant. A restaurant with the smallest average vector distance is then selected for recommendation. Alternatively, each member of the group might be requested to select a menu item from any of the restaurants in the restaurant database. The system then analyzes the recipe vectors of the chosen items, and selects a restaurant that best satisfies the menu items selected. If a particular menu item is not located in the selected restaurant's record 932, the system finds the closest substitute menu item and proposes it to the individual whose menu item was not located. The individual may accept the recommended item, or select a different item from the selected restaurant's menu. The system may further add the various menu items (e.g. 6 cheeseburgers, 2 fries, 5 cokes), and transmit the order via the Internet, fax, or other known communication means.

Detailed Description Text (112):

FIG. 25 is an exemplary GUI for receiving feedback from a user in regards to a recommended recipe. The user gives a rating 950 to each recommended dish 948 if it has been sampled by the user. For instance, if the user really liked a sampled dish, he or she may give it a rating of "8". On the other hand, if he or she just tolerated a dish, a rating of "3" is given. The ratings are submitted by selecting a "Submit" button 952.

Detailed Description Text (113):

The system utilizes the feedback received from the user to modify his or her food preference vector. According to one embodiment of the invention, a highly rated dish (e.g. dishes with rating of "7" or above) is merged into an existing cluster, as is described in further detail above. The ratings of the dishes are used to modify the values of the inclusive fields of the user's food preference vector. The amount by which a value is modified is proportional to the degree of dislike expressed by the user. For instance, if the protein field in the user's preference vector has a value of 30 (a value that is below average on a scale of 0 to 100), and the user gives a rating of "1" to a sampled dish, expressing a great dislike to the dish, the system might modify the protein field to a value of 90 (a value that is above average). This is done for every inclusive field in the user's preference vector. On the other hand, if the user only slightly disliked a dish, the inclusive field values may be modified only slightly, such

as modifying the protein field to a value of 35. For the lowly rated dishes (e.g. dishes with ratings of "2" or below) the system further creates negative cluster vectors to ensure that these dishes, or similar dishes, are not recommended in the future.

Detailed Description Text (115):

FIG. 27 is an exemplary GUI for the menu addition servlet 880 of FIG. 15. The GUI is available to a systems programmer for modifying recipes and recipe vectors, as well as adding new recipes directly into the recipe database 872. The systems programmer enters a new recipe by entering a recipe name 972, serving size 974, and preparation time 976 for the recipe. The programmer then selects the exclusive categories 978 to which the recipe belongs, and sets the exclusive fields of the corresponding recipe vector.

CLAIMS:

1. A computer-implemented method for recommending recipes catered to a particular user's preferences, the method comprising:

creating a user preference vector having a plurality of fields, each field corresponding to a chemical component contained in food;

obtaining a user preference for a particular type of dish;

analyzing the chemical components contained in the dish;

assigning values to the fields of the user preference vector corresponding to the chemical components contained in the dish, the values being based on an amount of each chemical component present in the dish;

retrieving a recipe from a menu recommendation database;

comparing the recipe with the user preference vector; and

selecting the recipe as a recommended recipe.

25. A computer system adapted to recommend recipes catered to a particular user's preferences, the system comprising:

a processor; and

a memory coupled to the processor and having program instructions stored therein, the processor being operable to execute the program instructions, the program instructions including:

creating a user preference vector having a plurality of fields, each field corresponding to a chemical component contained in food;

obtaining a user preference for a particular type of dish;

analyzing the chemical components contained in the dish;

assigning values to the fields of the user preference vector corresponding to the chemical components contained in the dish, the values being based on an amount of each chemical component present in the dish;

retrieving a recipe from a menu recommendation database;

comparing the recipe with the user preference vector; and

selecting the recipe as a recommended recipe.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)**End of Result Set**☐ [Generate Collection](#) [Print](#)

L6: Entry 1 of 1

File: USPT

Apr 9, 2002

DOCUMENT-IDENTIFIER: US 6370513 B1

TITLE: Method and apparatus for automated selection, organization, and recommendation of items

Brief Summary Text (6):

As another example, individuals are often faced with a decision as to what to cook for dinner. In today's environment most families consist of either two working spouses or a single working parent. As a consequence, at least one parent must come home from a long day at work and tackle the decision of what to make for dinner. Since the parent usually gets home from work around dinner time, the children and other family members are usually already hungry and tired. This situation often puts much stress on family members, in particular, the parent tasked with determining what meal to prepare (the primary care providing parent). The primary care providing parent is often pressured to prepare something quickly. Under this pressure, the parent opts most of the time for something simple and quick--if they have all the ingredients that they need to prepare the meal. If the parent does not have all the ingredients, he or she must make a trip to the store or figure out what they can make out of the ingredients at home. As a consequence of this situation, the family's menu is unimaginative. The family will therefore typically end up eating the same food over and over again, or the family will eat food from the same outside vendor (take-out).

Brief Summary Text (11):

If the system is to be used for recommending recipes for a family, it is desirable to have a system that will recommend recipes based on the tastes of all the family members. Just as a good maid or a stay-at-home parent, the system should learn and adapt to the family's food preferences, and make the meal selections accordingly. In addition, just as a maid or stay-at-home parent, the system would also track what meals the family has eaten in the past to avoid needless repetition.

Brief Summary Text (17):

The creation of product vectors for recommending dishes includes parsing an original recipe for its ingredients. These ingredients are mapped to the chemical components making up the ingredients. The value of each chemical component is then stored into the inclusive preference fields of the product vector.

Drawing Description Text (19):

FIG. 17 is an exemplary flow chart of a process for parsing and creating recipe vectors according to one embodiment of the present invention;

Drawing Description Text (24):

FIG. 22 is an exemplary GUI for displaying a weekly menu according to one embodiment of the present invention;

Drawing Description Text (25):

FIG. 23 is an illustration of an exemplary recipe recommended by the system;

Drawing Description Text (27):

FIG. 25 is an exemplary GUI for receiving feedback from a user in regards to a recommended recipe;

Drawing Description Text (28):

FIGS. 26A-26D are exemplary GUIs for adjusting the weight of chemical compositions appearing in a recipe;

Drawing Description Text (29):

FIG. 27 is an exemplary GUI for the menu addition servlet of FIG. 15; and

Drawing Description Text (30):

FIG. 28 is an exemplary GUI for adding or modifying ingredients of a recipe in accordance with one embodiment of the present invention.

Detailed Description Text (11):

FIG. 4 illustrates a user's food preference vector 75A used for recommending recipes to the user. The exclusive positions represent non-waivable preferences relating to the dishes that are recommended. For example, if an exclusive position 80A in the preference vector indicates that the user is vegetarian, the recipe selected must be a vegetarian recipe. The inclusive fields 90A indicate the user's degree of preference for certain types of foods or tastes.

Detailed Description Text (15):

Each product vector is further associated with additional information about the product also stored in the recommendation database 32, or alternatively, in a separate database. FIG. 5 is a conceptual layout diagram detailing the organization of product information in the recommendation database 32. As illustrated in FIG. 5, the products are categorized into broad categories 50, such as music, movies, recipes, books, and the like. The broad categories 50 are further divided into one or more sub-categories 52 for further categorizing the products. According to one embodiment of the invention, the sub-categories 52 are the exclusive fields of the product vector. Thus, a music category 910 is further divided into, for instance, Classical, Jazz, and Heavy Metal sub-categories 52. A recipes category is divided into Entree, Appetizer, Soup, Salad, and Dessert sub-categories. The Entree sub-category is further divided into Indian foods, Italian foods, Vegetarian foods, and the like. It is noted at this point, that other organization methods may also be utilized, such as organizing the products alphabetically within each broad category 50. Alternatively, the recipes category may be divided into eating type, dish type, meal type, diets, ethnicity, and ingredients.

Detailed Description Text (17):

A product record 54 in the recipes category, on the other hand, is headed by a dish name 64. The dish name is followed by the list of ingredients 66, preparation instructions 68, and graphic data of an image of the prepared dish. Alternatively, rather than maintaining all the information relating to a product within the local database 16, a product record 54 may simply contain a pointer to a web page stored in an offsite database. The web page will contain the product information that would otherwise be stored in the product record 54.

Detailed Description Text (46):

For certain items, it may be desirable to keep track of the recommended items actually sampled by the user, as well as when the sampling took place. For instance, in a recipe recommendation system, it may be desirable to keep a time factor attached to each sampled recipe to prevent recipes from being recommended too often. This helps to add variety to a user's menu. According to one embodiment of the invention, the time factor changes from zero to a value between zero and one. The system remembers when an item was last used by storing the date the item was sampled. The system also has a standard number of days after which an item can be used again. If the item has not been used within this time, the time factor for the item is set to one. Otherwise, the time factor is set to a fraction of one based on the number of days it was last used. For example, for a system where the standard number of days is set to ten, an item that has not been used in the past ten days will have one as its time factor. On the other hand, an item that was used only five days ago will have its time factor set to 0.5. According to this embodiment, the program asks in step 460 whether there are any more items in the sorted list of suitable items on which to perform a time factor calculation. If there are, the program, in step 470, multiplies the suitability weight of a current item in the list with a corresponding time factor. This process continues until the suitability weights of all the items in the sorted list have been recomputed based on their time factors. The list is then rearranged in step 480 based on the new calculations.

Detailed Description Text (47):

In step 490, the program recommends the top items from the sorted list. The actual number of items recommended is predetermined by either the user or systems programmer. According to one embodiment of the invention, the top seven items are recommended as the menu for the week.

Detailed Description Text (48):

In a recipe recommendation system, the system may also recommend wines which may most likely complement a recommended dish. FIG. 8 is a flow diagram of a computer program for recommending wines, in accordance with one embodiment of the invention. According to this embodiment, the system maintains a preset list of dishes along with the name of wines that best complement each dish. In making a wine recommendation, the program starts, and in step 60, inquires whether there are more dishes to examine in the preset list of wine dishes. If the answer is yes, the program proceeds to retrieve the product vector of a wine dish. This product vector is preferably stored in the recommendation database 32 (FIG. 2). In step 64, the program computes the vector distance between the wine dish's product vector, and the product vector of a dish that is recommended to the individual. In step 66, the program places the wine dish into a sorted list according to its vector distance. If all the wine dishes in the list have been examined and placed according to their vector distances, the program next selects, in step 68, a wine dish with the smallest vector distance to the dish that is recommended to the user. In step 70, the program retrieves the wine name associated with the selected wine dish. The program, in step 72, then recommends the retrieved wine to the user, as complementing the dish which is recommended.

Detailed Description Text (49):

As an individual uses the system over time, the system learns and adapts to the user's preferences. FIG. 9 illustrates the initial learning and adaptation process according to one embodiment of the invention. In step 230, the program proposes an initial list of items to the user. In step 232, the user either accepts or rejects the recommended items. If the items are rejected, the program in step 234 asks feedback questions to ascertain why the items were rejected. Similarly, if the items are accepted, the program in step 236, asks questions to ascertain why each item was accepted. In step 238, the user preference vector 75 is updated, if necessary, and used for future choices. For instance, if a recommended recipe was rejected because it was too spicy, the value in the spiciness field of the user preference vector might be decreased. As a user uses the system over time, the recommendations become more and more accurate and feedback from the user becomes less and less required. In this case, the system may no longer require this type of initial feedback from the user.

Detailed Description Text (50):

FIG. 10 illustrates the process of obtaining feedback from the user for further updating a preference vector after the user has actually sampled a recommended item. The computer program begins and in step 300 asks a question about a particular attribute of the sampled item. Examples of such questions for a system recommending CDs include: "What rating would you give to the recommended CD?"; "What did you like most? Tempo? Instruments? Lyrics?"; and "What did you dislike about the CD?" Examples of questions for a system recommending recipes might include: "What rating would you give to the recommended meal?"; "What did you like most? Taste? Amount of preparation required?"; "Which ingredients did you like or dislike?"; "How should the meal have been changed? Less salty? Lighter?"; and "Was there enough food?"

Detailed Description Text (51):

Based on the user response, the program in step 310 determines whether there was too much of the inquired attribute in the recommended item. If there was, the value in the user's preference vector corresponding to that attribute is decreased in step 320. Likewise, if the program determines in step 320 that there was too little of the inquired attribute in the recommended item, the value in the user's preference vector corresponding to that attribute is increased in step 340. For example, if the songs in a recommended CD were too slow, the value of the field in the user preference vector 75 representing the individual's preference for slow music would be decreased below the value in the corresponding field of the CD's product vector. If a recommended recipe was too salty, the value of the field in the family food preference vector representing the family's preferred saltiness would be decreased below the value in the corresponding field of the recipe product vector.

Detailed Description Text (65):

In a recipe recommendation system, the individual user's personal computer 14, set-top box 18, or HPC 20 (FIG. 1) optionally includes an inventory control subsystem which keeps track of ingredients used for meal preparations. According to this embodiment, the personal computer 14, set-top box 18, or HPC hosts a home inventory database storing an inventory table of ingredients available at the user's home. Each entry in the table specifies a UPC code for the ingredient, the amount available, and the expiration date. New items can be automatically or

manually added to the database. For instance, every time a grocery item is ordered via the Internet, the inventory control subsystem automatically inserts the item ordered into the inventory table. Alternatively, the update may be made when the goods are actually delivered to the user. In this scenario, the individual uses a bar coder reader to manually scan the UPC code on the goods delivered. The information may also be keyed into the subsystem via the keyboard or touch screen display.

Detailed Description Text (66):

Update to the inventory table is made each time a recommended recipe is selected for cooking. For example, if a recipe calls for two eggs, the subsystem subtracts two from the total number of eggs listed in the inventory table. This method of keeping inventory requires some feedback from the family. For instance, if a user decides to discard a product, he or she should notify the inventory subsystem so that it can be subtracted from the home inventory database. Such notification may be accomplished by scanning the product label via the barcode reader, and keying-in the amount used.

Detailed Description Text (81):

Alternative Recipe Recommendation System

Detailed Description Text (82):

FIG. 15 is a block diagram of a configuration of an alternative network server or platform computer 10 of FIGS. 1 and 2, for specifically recommending recipes. According to the illustrated embodiment, the system hosts a user preference database 870 storing user preference vectors that map the food tastes of the individual users of the system. FIG. 16 illustrates an exemplary individual's food preference vector 75B. The vector is divided into exclusive fields 80B and inclusive fields 90B. The exclusive fields 90A depict specific categories of foods to exclude in making a recommendation.

Detailed Description Text (84):

Referring back to FIG. 15, a recipe database 872 stores a recipe vector for each dish capable of being recommended by the system. Each recipe vector is identified by the name of the dish. According to one embodiment of the invention, the recipe vector includes the same exclusive fields and inclusive fields as a user's food preference vector 75B (FIG. 16). Thus, a recipe vector will have the appropriate exclusive fields set based on the food category it belongs, and the appropriate inclusive fields set based on the amount of chemical components contained in the recipe.

Detailed Description Text (85):

In an alternative embodiment, the recipe vector includes only the inclusive fields depicting the chemical components of the dish. According to this embodiment, the recipe vectors are stored under appropriate categories based on the type of dish being represented. Each category is associated with an exclusive field of a user's food preference vector. Thus, a recipe vector for a vegetarian dish is stored under a Vegetarian category while a recipe vector for a Breakfast dish is stored under a Breakfast category. Furthermore, a single recipe vector may belong to multiple categories.

Detailed Description Text (86):

Recipe vectors are created with the aid of a parser 874 which is in communication with an original recipes directory 876 and a chemical database 878. The parser 874 takes a recipe in the original recipes directory 876, parses out the ingredients in the recipe, and maps the ingredients to the chemicals in the chemical database 878, as is discussed in further detail below. A parsed recipe may be modified via a menu addition servlet 880. The servlet, moreover, allows recipes to be entered directly into the recipe database 872 without invoking the parser.

Detailed Description Text (88):

FIG. 17 is a flow diagram of an exemplary parsing and recipe vector creation process according to one embodiment of the invention. The computer program starts by taking an original recipe from the original recipes directory 876, and in step 954, inquires whether there are any more ingredients to parse. If the answer is yes, the program parses out an ingredient in step 956. The program also asks in step 958 whether the parsed ingredient exists in the chemical database 878. If it does, the system maps the chemical compositions making up the ingredient to an ingredient vector. In doing so, the system searches the chemical database for a record

corresponding to the ingredient. If the record is found, the system sets the values of the ingredient vector according to the chemical values stored in the located record. The system then multiplies the values in the ingredient vector with the weight/amount of the ingredient called for in the recipe. In addition, the system might multiply each field in the ingredient vector by the weight (scaling coefficient) assigned to the field. Chemical compositions which make greater contributions to an ingredient's taste and attribute are given higher weights than those that do not have much effect on neither taste nor attribute. For instance proteins and sugars are given a maximum possible weight (e.g. 100 in a scale of 0-100), while energy and calcium are given low weights (e.g. 0.1 in a scale of 0-100).

Detailed Description Text (89):

If the program does not find an ingredient in the chemical database, a substitute ingredient that is found in the database is used in its place. The substitution may be done manually by a systems programmer via the menu addition servlet 880. Alternatively, the system might select a key phrase in the ingredient's name, and find an ingredient in the recipe database that includes the selected key phrase.

Detailed Description Text (90):

After all the ingredients of the given recipe have been parsed, the computer program, in step 964, adds the chemicals found in the various ingredients by performing a vector addition of all the ingredient vectors. The resultant vector is saved as a recipe vector, and in step 966, is normalized for hundred grams of the entire recipe. In step 968, the exclusive fields of the recipe vector are set via a systems programmer, and the process ends.

Detailed Description Text (91):

The system also hosts a USDA servlet 882 which allows addition of ingredients and their corresponding chemical compositions, into the chemical database 878. Thus, if an ingredient in a recipe being parsed is not found in the chemical database, the ingredient and its chemical composition may be added to the database. Alternatively, the closest match to the ingredient missing from the database may be used to map the chemicals.

Detailed Description Text (92):

A recommendation engine 884 in conjunction with a search servlet acts to find recipes that will cater to an individual's tastes. According to a currently preferred embodiment, the recommendation engine computes the vector distance between the user's food preference vector and each recipe vector to find the dishes to recommend.

Detailed Description Text (94):

FIG. 19 is an exemplary GUI for allowing entry of preference information from the user for initializing the user's food preference vector. The user may use the GUI to enter his or her favorite dish 900, and select a find dishes button 902. If the name of the dish exists in the recipe database, the dish typed-in by the user is accepted. Otherwise, if the system cannot find an exact match, the system displays a list of other comparable dishes for user selection. In its most general form, this is accomplished by finding dishes with names that partly match the dish name specified by the user. For instance, the user may type-in "Spaghetti" as his or her favorite dish. If the recipe database 872 (FIG. 15) does not contain a dish simply called "Spaghetti" but does contain dishes with the word "Spaghetti" such as "Spaghetti and meatballs", "Seafood spaghetti", and "Spaghetti and white clam sauce", the list of such dishes are displayed to the user for his or her selection.

Detailed Description Text (95):

The system further allows the user to rate up to five other dishes that the user likes 904 to get better knowledge of the user's tastes. The user accesses a list of dishes stored in the recipes database 872 by selecting a down-arrow button 906, and further selecting a dish from a resultant pull-down menu of dishes.

Detailed Description Text (97):

Upon the completion of the form illustrated in FIG. 19, the user selects a "Submit ratings" button, and submits the responses to the system's network server or platform computer 10 (FIG. 1). The system then proceeds to create and/or update the user's food preference vector. If the user has entered a favorite dish 900, the system retrieves the recipe vector for the particular dish, and copies the values of the recipe's inclusive fields, into the inclusive fields of the user's preference vector. The exclusive fields are set according to the responses to questions

posed by the system regarding to the user's diet 906 and allergies 908.

Detailed Description Text (98):

According to a currently preferred embodiment, the system creates food preference vectors for the other five dishes the user has rated to be to his or her liking 904. Each preference vector acts as a cluster vector. As the user uses the system and indicates other dishes to be his or her favorite, the recipe vectors for those dishes are merged into a cluster with a smallest vector distance to the new favorite dish. Cluster vectors and various alternatives in creating cluster vectors is discussed above in greater detail.

Detailed Description Text (99):

After one or more preference vectors have been created, the system may now make recommendations on recipes that will cater to the individual's tastes. In addition, the system may also make recommendations that cater to a group of individuals who have registered into the system. FIG. 20 is an exemplary group setup GUI. In accordance with one embodiment of the invention, the user may create a new group, or add himself or herself to an existing group, by selecting a group setup option 912. When a user first registers onto the system, a new group is created with the individual as the initial member and creator of the group. A creator of the group is given special privileges, such as the ability to delete the group, and add or delete members to the group.

Detailed Description Text (100):

In making a recommendation for a group, the system selects a recipe in the recipe database and computes the vector distance to the nearest product cluster of each member in the group. The average distance to the recipe is then calculated by adding the vector distances to the recipe for all the members, and dividing the total distance by the number of members in the group. This is done for each recipe in the recipe database. The system then recommends the recipe with the smallest average vector distance.

Detailed Description Text (102):

A user may view his or her weekly menu by selecting a weekly menu option 914, as illustrated in FIG. 22. The user may also view recommendations for a different number of days (e.g. the next two weeks) by entering a desired number in a "Number of days" 916 field, and selecting a "find dishes" button 924. For a day specified as a cook-in day, the system displays recommendations of specified type of dishes (e.g. soup, salad, entree, etc.). A "Show Recipe" button 924 next to the recommended dish allows the user to view a picture of the prepared dish, the dish ingredients, and preparation instructions.

Detailed Description Text (103):

FIG. 23 illustrates an exemplary recipe displayed upon selection of the "Show Recipe" button 924. The ingredients necessary may be added to a shopping list by selecting an "Add to Shopping List" button 926. Furthermore, the recipe may be e-mailed 928 and/or faxed 930 to the user, if so desired. According to one embodiment of the invention, multimedia presentations are used in conjunction with the written instructions to instruct a family member in how to prepare the recommended dish. The multimedia presentation will typically include a video/audio presentation. In other cases, references to cookbooks will be made for the user to look up the instructions in a specified cookbook.

Detailed Description Text (106):

Each restaurant record 932 also includes an information storage area with a list of dishes 942 offered by the restaurant. In a currently preferred embodiment, each dish is associated with a recipe vector in the recipe database 872 (FIG. 15). Associated with each listed dish 942 are the dates 944 in which the dish is offered. For example, the dish may be offered everyday, or on certain days of the week (e.g. Sundays). The dish may also be offered for a limited period of time (e.g. June 1-June 28). Also associated with each dish are comments 946 related to the dish, if such was provided by the restaurant.

Detailed Description Text (107):

In recommending a restaurant to a user, the system analyzes the dishes offered by each restaurant, and computes the vector distance between the user's food preference vector and a restaurant's recipe vector. The restaurant with a dish with the smallest vector distance is then recommended.

Detailed Description Text (108):

If a restaurant is to be recommended to a group of individuals, the system calculates the vector distance to the dishes of a particular restaurant, and calculates the average vector distance for that restaurant. A restaurant with the smallest average vector distance is then selected for recommendation. Alternatively, each member of the group might be requested to select a menu item from any of the restaurants in the restaurant database. The system then analyzes the recipe vectors of the chosen items, and selects a restaurant that best satisfies the menu items selected. If a particular menu item is not located in the selected restaurant's record 932, the system finds the closest substitute menu item and proposes it to the individual whose menu item was not located. The individual may accept the recommended item, or select a different item from the selected restaurant's menu. The system may further add the various menu items (e.g. 6 cheeseburgers, 2 fries, 5 cokes), and transmit the order via the Internet, fax, or other known communication means.

Detailed Description Text (109):

Referring back to FIG. 22, a user may request that the recommendations made for the entire week, whether it be a particular dish and/or restaurant recommendation, be e-mailed and/or faxed to the user. The user makes these requests by selecting a "fax recipes" button 920 or an "email recipes" button 922, respectively. Alternatively, the system automatically e-mails or faxes the recommendations for the week, at the beginning of each week. In this way, the user need not revisit the system to get the recommendations once he or she is registered.

Detailed Description Text (110):

In addition, an "Add all to Shopping Cart" option 920 causes the system to prepare a shopping list of all ingredients necessary for preparing the cook-in meals for the week (or an otherwise specified number of days). In doing so, the system adds the recommended quantities of ingredients required in more than one recipe, rather than listing the same ingredient in multiple locations of the list. For example, if the recipes recommended for day one and day three both require a cup of sugar, the system places two cups of sugar into the shopping list instead of placing a cup of sugar in two separate listings. Furthermore, the system determines whether an ingredient is offered by one of the sponsors of the system. If this is the case, the sponsor's brand name is suggested for the ingredient. For instance, if one of the ingredients to be inserted into the shopping list is cream cheese, and one of the sponsors of the system is Kraft Foods, Inc., the system would place Philadelphia.RTM. cream cheese into the user's shopping list. If the system has access to an inventory database, as is described in further detail above, the system places an ingredient into the shopping list if the user is running low on the ingredient.

Detailed Description Text (111):

The system further allows a user to search for dishes which taste similar to a dish entered. A user does so by entering a dish and selecting a "Search" button 925. The system then searches the recipe database 872 and displays a list of dishes with the smallest vector distance. A user may similarly find dishes which include specified types of ingredients or find dishes that exclude specified types of ingredients.

Detailed Description Text (112):

FIG. 25 is an exemplary GUI for receiving feedback from a user in regards to a recommended recipe. The user gives a rating 950 to each recommended dish 948 if it has been sampled by the user. For instance, if the user really liked a sampled dish, he or she may give it a rating of "8". On the other hand, if he or she just tolerated a dish, a rating of "3" is given. The ratings are submitted by selecting a "Submit" button 952.

Detailed Description Text (113):

The system utilizes the feedback received from the user to modify his or her food preference vector. According to one embodiment of the invention, a highly rated dish (e.g. dishes with rating of "7" or above) is merged into an existing cluster, as is described in further detail above. The ratings of the dishes are used to modify the values of the inclusive fields of the user's food preference vector. The amount by which a value is modified is proportional to the degree of dislike expressed by the user. For instance, if the protein field in the user's preference vector has a value of 30 (a value that is below average on a scale of 0 to 100), and the user gives a rating of "1" to a sampled dish, expressing a great dislike to the dish, the system might modify the protein field to a value of 90 (a value that is above average). This is done for every inclusive field in the user's preference vector. On the other hand, if the user

only slightly disliked a dish, the inclusive field values may be modified only slightly, such as modifying the protein field to a value of 35. For the lowly rated dishes (e.g. dishes with ratings of "2" or below) the system further creates negative cluster vectors to ensure that these dishes, or similar dishes, are not recommended in the future.

Detailed Description Text (114):

FIGS. 26A-26D are exemplary GUIs for adjusting the weights (scaling coefficients) 960 of the chemical compositions 962 appearing in the inclusive fields of a recipe vector or a user preference vector. The weights 960 are preferably set based on the contribution of each chemical composition to a dish's taste or attribute. In creating a recipe vector for a particular dish, the system multiplies the weight of a chemical composition with the amount of the chemical present in the dish. Thus, if a dish contains six grams of protein, and the protein attribute is given a weight of 100, the value in the protein field before normalization would be 600 (6.times.100). An individual user of the system or a systems programmer may increase or decrease the weight factors by selecting a "+" icon or a "-" icon, respectively. Furthermore, a user may view the amount of each chemical composition 964 in a particular dish, by entering the name of a desired dish 966 found in the recipe database. If the user enters a name of a second dish 968, the system displays the chemical compositions in the second dish, as well as the vector distance 970 between the first dish 966 and the second dish 968.

Detailed Description Text (115):

FIG. 27 is an exemplary GUI for the menu addition servlet 880 of FIG. 15. The GUI is available to a systems programmer for modifying recipes and recipe vectors, as well as adding new recipes directly into the recipe database 872. The systems programmer enters a new recipe by entering a recipe name 972, serving size 974, and preparation time 976 for the recipe. The programmer then selects the exclusive categories 978 to which the recipe belongs, and sets the exclusive fields of the corresponding recipe vector.

Detailed Description Text (116):

A "Show Ingredients" button 980 allows the programmer to enter the ingredients for the new recipe. FIG. 28 is an exemplary GUI for entering ingredients upon selection of the "Show Ingredients" button 980. After entry of the ingredients, the user selects a "Map" button 982 for mapping the ingredients to chemicals in the chemical database 878 (FIG. 15), and setting the inclusive fields of the corresponding vector.

Detailed Description Text (117):

If the systems programmer desires to modify an existing recipe, he or she enters the desired recipe name 972 and selects a "Search" button 984. The programmer may then make modifications to the exclusive categories 978, or add/modify ingredients by selecting the "Show Ingredients" button 980. The modified recipe is then re-mapped by selecting the "Map" button 982 of FIG. 28.

Detailed Description Text (118):

While the invention has been described with respect to particular illustrated embodiments, those skilled in the art and technology to which the invention pertains will have no difficulty devising variations which in no way depart from the invention. For instance, the exclusive and inclusive fields may be maintained as separate vectors. Furthermore, the method of automating the creation of recipe vectors may be extended to automate the creation of other product vectors. For example, in creating product vectors for musical pieces, analysis of the musical pieces may be performed via an automated DSP (digital signal processing) algorithm. This would allow the automatic detection of the kinds of instruments involved as well as other musical attributes necessary to create the product vectors. For a painting recommendation system, color and texture analysis may be correlated to attributes present in paintings to automatically create a product vector for a particular painting. Furthermore, the described system for recommending items may be extended to other types of knowledge-based selection systems where recommendations are made based on the knowledge of a user's preference. Accordingly, the present invention is not limited to the specific embodiments described above, but rather as defined by the scope of the appended claims and their equivalents.

CLAIMS:

1. A computer-implemented method for recommending recipes catered to a particular user's preferences, the method comprising:

creating a user preference vector having a plurality of fields, each field corresponding to a chemical component contained in food;

obtaining a user preference for a particular type of dish;

analyzing the chemical components contained in the dish;

assigning values to the fields of the user preference vector corresponding to the chemical components contained in the dish, the values being based on an amount of each chemical component present in the dish;

retrieving a recipe from a menu recommendation database;

comparing the recipe with the user preference vector; and

selecting the recipe as a recommended recipe.

2. The method of claim 1 further comprising creating a recipe vector for the recipe, the recipe vector having a plurality of fields, the fields of the recipe vector corresponding to the fields of the user preference vector.

3. The method of claim 2, wherein the comparing includes calculating a vector distance between the user preference vector and the recipe vector.

4. The method of claim 2, wherein the creating the recipe vector further comprises:

parsing the recipe for ingredients contained in the recipe;

searching for the ingredients in a chemical database;

retrieving chemical components found in each of the ingredients; and

assigning values to the fields of the recipe vector corresponding to the retrieved chemical components, the values being based on a total amount of each retrieved chemical component.

5. The method of claim 1 further comprising:

obtaining user feedback about the recommended recipe; and

modifying the value of at least one of the fields of the user preference vector based on the user feedback.

6. The method of claim 5, wherein the obtaining user feedback further comprises obtaining a user rating for the recommended recipe, and the modifying the value further comprises:

analyzing the chemical components contained in the recipe;

locating the fields of the user preference vector corresponding to the chemical components contained in the recipe; and

changing the values of the retrieved fields, the change of value being proportional to the rating provided by the user.

8. The method of claim 1 further comprising:

maintaining a list of products offered by a particular vendor in a retailer inventory database;

parsing the recommend recipe for specific ingredients;

searching the retailer inventory database for the specific ingredients; and

inserting matching products offered by the particular vendor into a shopping list.

9. The method of claim 1, wherein each recipe in the recipe database is associated with a time factor, and the method further comprises analyzing the time factor associated with the retrieved recipe.

10. In a computer for recommending recipes catered to a particular user's preference, each recipe being represented by a recipe vector having a plurality of fields, each field corresponding to a chemical component contained in food, a method for creating the recipe vector for a particular recipe comprising:

maintaining a chemical database storing a list ingredients and chemical components found in each of the ingredients;

parsing the particular recipe for ingredients contained in the particular recipe;

searching for the parsed ingredients in the chemical database;

retrieving the chemical components found in each of the parsed ingredients; and

assigning values to the fields of the recipe vector corresponding to the retrieved chemical components, the values being based on a total amount of each retrieved chemical component, the recipe vector being used for comparing against a user preference vector for determining whether the particular recipe may be recommended to the user.

12. The method of claim 10, further comprising:

creating an ingredient vector for each of the parsed ingredients, the ingredient vector having a plurality of fields, the fields of the ingredient vector corresponding to the fields of the recipe vector; and

assigning values to the fields of the ingredient vector, the values being based on an amount of each retrieved chemical component found in the parsed ingredient.

14. A computer system for recommending recipes catered to a particular user's preferences, the system comprising:

means for creating a user preference vector having a plurality of fields, each field corresponding to a chemical component contained in food;

means for obtaining user preference for a particular type of dish;

means for analyzing the chemical components contained in the dish;

means for assigning values to the fields of the user preference vector corresponding to the chemical components contained in the dish;

means for storing recipes for being recommended to the user; and

means for comparing the recipes with the user preference vector;

means for selecting one or more recipes as a recommended recipe.

15. The system of claim 14 further comprising means for creating a recipe vector for the recipe, the recipe vector having a plurality of fields, the fields of the recipe vector corresponding to the fields of the user preference vector.

16. The system of claim 15, wherein the means for creating the recipe vector further comprises:

means for parsing the recipe for ingredients contained in the recipe;

means for searching for the ingredients in a chemical database;

means for retrieving chemical components found in each of the ingredients; and

means for assigning values to the fields of the recipe vector corresponding to the retrieved chemical components.

17. The system of claim 14 further comprising:

means for obtaining user feedback about the recommended recipe; and

means for modifying the value of at least one of the fields of the user preference vector based on the user feedback.

18. The system of claim 17, wherein the means for obtaining user feedback further comprises means for obtaining a user rating for the recommended recipe, and the means for modifying the value further comprises:

means for analyzing the chemical components contained in the recipe;

means for locating the fields of the user preference vector corresponding to the chemical components contained in the recipe; and

means for changing the values of the retrieved fields.

20. The system of claim 14 further comprising:

means for maintaining a list of products offered by a particular vendor;

means for parsing the recommend recipe for specific ingredients;

means for searching for the specific ingredients; and

means for inserting matching products offered by the particular vendor into a shopping list.

21. The system of claim 14 further comprising means for analyzing a time factor associated with the retrieved recipe.

22. computer system for creating a recipe vector for recommending a recipe catered to a particular user's preferences, the recipe vector having a plurality of fields, each field corresponding to a chemical component contained in food, the system comprising:

means for storing a list of ingredients and chemical components found in each of the ingredients;

means for obtaining ingredients contained in the particular recipe;

means for searching for the obtained ingredients in the chemical database;

means for retrieving the chemical components found in each of the obtained ingredients; and

means for assigning values to the fields of the recipe vector corresponding the retrieved chemical components the recipe vector being used for comparing against a user preference vector for determining whether the particular recipe may be recommended to the user.

24. The system of claim 22 further comprising:

means for creating an ingredient vector for each of the obtained ingredients, the ingredient vector having a plurality of fields, the fields of the ingredient vector corresponding to the fields of the recipe vector; and

means for assigning values to the fields of the ingredient vector.

25. A computer system adapted to recommend recipes catered to a particular user's preferences,

the system comprising:

a processor; and

a memory coupled to the processor and having program instructions stored therein, the processor being operable to execute the program instructions, the program instructions including:

creating a user preference vector having a plurality of fields, each field corresponding to a chemical component contained in food;

obtaining a user preference for a particular type of dish;

analyzing the chemical components contained in the dish;

assigning values to the fields of the user preference vector corresponding to the chemical components contained in the dish, the values being based on an amount of each chemical component present in the dish;

retrieving a recipe from a menu recommendation database;

comparing the recipe with the user preference vector; and

selecting the recipe as a recommended recipe.

26. The computer system of claim 25, wherein the program instructions further include creating a recipe vector for the recipe, the recipe vector having a plurality of fields, the fields of the recipe vector corresponding to the fields of the user preference vector.

27. The computer system of claim 26, wherein the comparing includes calculating a vector distance between the user preference vector and the recipe vector.

28. The computer system of claim 26, wherein the creating the recipe vector further comprises:

parsing the recipe for ingredients contained in the recipe;

searching for the ingredients in a chemical database;

retrieving chemical components found in each of the ingredients; and

assigning values to the fields of the recipe vector corresponding to the retrieved chemical components, the values being based on a total amount of each retrieved chemical component.

29. The computer system of claim 25, wherein the program instructions further include:

obtaining user feedback about the recommended recipe; and

modifying the value of at least one of the fields of the user preference vector based on the user feedback.

30. The computer system of claim 29, wherein the obtaining user feedback further comprises obtaining a user rating for the recommended recipe, and the modifying the value further comprises:

analyzing the chemical components contained in the recipe;

locating the fields of the user preference vector corresponding to the chemical components contained in the recipe; and

changing the values of the retrieved fields, the change of value being proportional to the rating provided by the user.

32. The computer system of claim 25, wherein the program instructions further include:

maintaining a list of products offered by a particular vendor in a retailer inventory database;

parsing the recommend recipe for specific ingredients;

searching the retailer inventory database for the specific ingredients; and

inserting matching products offered by the particular vendor into a shopping list.

33. The computer system of claim 25, wherein each recipe in the recipe database is associated with a time factor, and the method further comprises analyzing the time factor associated with the retrieved recipe.

34. A computer system for creating a recipe vector for recommending a recipe catered to a particular user's preferences, the recipe vector having a plurality of fields, each field corresponding to a chemical component contained in food, the system comprising:

a memory storing a chemical database including a list of ingredients and chemical components found in each of the ingredients, the memory having program instructions stored therein; and

a processor coupled to the memory, the processor being operable to execute the program instructions, the program instructions including:

parsing the particular recipe for ingredients contained in the particular recipe;

searching for the parsed ingredients in the chemical database;

retrieving the chemical components found in each of the parsed ingredients; and

assigning values to the fields of the recipe vector corresponding to the retrieved chemical components, the values being based on a total amount of each retrieved chemical component, the recipe vector being used for comparing against a user preference vector determining whether the particular recipe may be recommended to the user.

36. The computer system of claim 34, wherein the program instruction further include:

creating an ingredient vector for each of the parsed ingredients, the ingredient vector having a plurality of fields, the fields of the ingredient vector corresponding to the fields of the recipe vector; and

assigning values to the fields of the ingredient vector, the values being based on an amount of each retrieved chemical component found in the parsed ingredient.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

Freeform Search

Database:	<div style="border: 1px solid black; padding: 2px;"> US Pre-Grant Publication Full-Text Database US Patents Full-Text Database US OCR Full-Text Database EPO Abstracts Database JPO Abstracts Database Derwent World Patents Index IBM Technical Disclosure Bulletins </div>
Term:	<div style="border: 1px solid black; padding: 2px;"> L5 and (dish near rating) </div>
Display:	<div style="border: 1px solid black; padding: 2px;">100</div> Documents in Display Format: <div style="border: 1px solid black; padding: 2px;">-</div> Starting with Number <div style="border: 1px solid black; padding: 2px;">1</div>
Generate: <input type="radio"/> Hit List <input checked="" type="radio"/> Hit Count <input type="radio"/> Side by Side <input type="radio"/> Image	

Search

Clear

Interrupt

Search History

DATE: Wednesday, March 16, 2005 [Printable Copy](#) [Create Case](#)

<u>Set Name</u> side by side	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u> result set
<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>			
<u>L6</u>	L5 and (dish near rating)	1	<u>L6</u>
<u>L5</u>	L4 and recipe\$	16	<u>L5</u>
<u>L4</u>	L1 and menu	32	<u>L4</u>
<u>L3</u>	L2 and (dish near rating)	0	<u>L3</u>
<u>L2</u>	L1 and (ingredient near information)	12	<u>L2</u>
<u>L1</u>	ingredient near dish\$	171	<u>L1</u>

END OF SEARCH HISTORY